

SIXTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (5 points)

What is the value of each of the following expressions? (Note that `negative?` is a function that takes a number and returns true if it's less than zero.)

```
(define square
  (lambda (n)
    (* n n)))
```

(a) `(filter negative? (list 5 4 3 2 1 0 -1 -2 -3 -4 -5))`

(b) `(filter (lambda (L) (not (empty? L)))
 (list (list 'a 'b) empty (list 'b 'c 'c) (rest (list 'd))))`

(c) `(map square (list 2 3 4))`

(d) `(map (lambda (n)
 (cond
 ((negative? n) 0)
 (else (square n))))
 (list 3 -3 2 -2 1 -1 0))`

(e) `(foldr + 0 (list 10 20 30 40))`

(f) `(foldr (lambda (a b) (and a b)) true (map negative? (list 1 -2 3 -4 5)))`

Problem 2 (15 points)

Suppose we have a list called BL of books defined as follows:

```
(define-struct book (title author genre price sales))
```

where title and author are strings, genre is a symbol (e.g., `'fiction` or `'travel`) representing the category of the book, price is a number representing the price of one copy, and sales is a number representing the number of copies of this book that have been sold.

(a) For each of the following expressions, describe in one English phrase what value it returns. Don't just say, "It does a foldr of plus and zero and ..."; give a description of what the expression *means*, something you could put in a software catalog so that a prospective buyer could find what he or she wanted. Assume that BL is a list of books and that `member?` is a function that takes an item and a list and returns true if the item occurs on the list.

(a.1) `(map book-title (filter (lambda (B) (symbol=? (book-genre B) 'cookbook)) BL))`

(a.2) `(map book-author
 (filter (lambda (B) (member? (book-genre B) (list 'home 'gardening 'cookbook)))
 (filter (lambda (B) (>= (* (book-price B) (book-sales B)) 1000000)) BL)))`

(b) Using `map`, `filter`, and/or `foldr`, define the following function without using explicit recursion.

```
;; average-price-by-genre: symbol (listof book) -> number  
;; Return the average price of all books in the specified genre.
```